
Software Requirements Specification (SRS)

Customer Feedback System

---

1. Introduction

1.1 Purpose

The Customer Feedback System is a web-based application developed to allow customers to share product feedback, rate their experiences, and enable administrators to manage reviews and products effectively.

It aims to improve product quality and customer satisfaction by collecting, storing, and analyzing user feedback.

1.2 Scope

The system provides:

- User registration and login
- Product catalog viewing and management
- Shopping cart and order features
- Feedback and rating submission
- Admin dashboard for managing products and feedback

1.3 Technologies Used

Layer	Technologies
Frontend	HTML, CSS, JavaScript, Bootstrap
Backend	Java, Spring Boot Framework
Database	MySQL
Security	JWT Authentication, BCrypt Encryption
Build Tool	Maven
Architecture	3-Tier (Frontend, Backend, Database)

---

2. Overall Description

2.1 Product Perspective

The Customer Feedback System is a standalone web application that connects the frontend (UI) with the backend REST APIs using Spring Boot and MySQL for data storage.

2.2 User Classes and Characteristics

User	Description
Customer	Can register, log in, view products, place orders, and provide feedback.
Administrator	Can manage products, view all feedback, and respond to customer comments.

2.3 Operating Environment

- Server: Supports Java 17+ (Spring Boot runtime)
- Database: MySQL 8.0 or higher
- Client: Modern web browsers (Chrome, Firefox, Safari, Edge)
- Network: Internet connection (HTTP/HTTPS)

---

3. Functional Requirements

3.1 User Management

- FR-1: Customers can register with name, email, and password.
- FR-2: Users can log in securely with JWT authentication.
- FR-3: Passwords are encrypted using BCrypt.
- FR-4: Admins and Customers have different access privileges.

3.2 Product Management

- FR-5: Admins can add, update, or delete products.
- FR-6: Customers can view all available products.
- FR-7: Products display name, price, description, and image.
- FR-8: Customers can search or filter products.

3.3 Shopping Cart and Order Management

- FR-9: Customers can add or remove products from the cart.
- FR-10: Customers can place orders from their cart.
- FR-11: The system generates an order confirmation message.

3.4 Feedback System

- FR-12: Customers can submit ratings (1–5 stars) and written comments.
- FR-13: Only customers who purchased a product can give feedback.
- FR-14: Admins can view and respond to customer feedback.
- FR-15: Average rating and customer comments are visible to all users.

3.5 Admin Dashboard

- FR-16: Admins can monitor all feedback and manage product listings.
- FR-17: Admins can reply to user comments or mark them as reviewed.

---

4. Non-Functional Requirements

4.1 Performance Requirements

- NFR-1: Average web page load time ≤ 3 seconds.
- NFR-2: REST API responses ≤ 2 seconds.
- NFR-3: Must support 100+ concurrent users.

4.2 Security Requirements

- NFR-4: Passwords hashed with BCrypt.
- NFR-5: JWT tokens for secure login sessions (24-hour validity).
- NFR-6: Prevent SQL Injection, XSS, and CSRF attacks through input validation.

4.3 Reliability Requirements

- NFR-7: System uptime should be 99.5%.
- NFR-8: Application should handle unexpected server or DB failures gracefully.

4.4 Usability Requirements

- NFR-9: UI must be simple, responsive, and easy to use.
- NFR-10: The design must adapt to desktops, tablets, and mobile screens using Bootstrap.

4.5 Maintainability Requirements

- NFR-11: Code must follow Java and Spring Boot standards.
- NFR-12: Include clear documentation for code, API endpoints, and database schema.

4.6 Compatibility Requirements

- NFR-13: Compatible with Chrome, Firefox, Safari, and Edge browsers.
- NFR-14: Works across Windows, Linux, and macOS operating systems.

---

5. External Interface Requirements

5.1 User Interface

- Built using HTML, CSS, JavaScript, and Bootstrap.
- Includes:
  - Navigation bar with Login, Products, Feedback, and Cart links
  - Responsive layout with cards for product display
  - Separate panels for Admin and Customer

5.2 Software Interfaces

- Backend: Spring Boot REST APIs
- Database: MySQL connected via JPA/Hibernate
- Authentication: JWT tokens
- Frontend: Communicates with backend via REST endpoints returning JSON data

5.3 Communication Interfaces

- Protocol: HTTP/HTTPS
- Data Format: JSON
- Secure Communication: HTTPS encryption for sensitive data

---

6. Database Design

6.1 Main Tables

Users

Field	Type	Description
id	BIGINT	Primary Key
name	VARCHAR(100)	Full Name
email	VARCHAR(100)	Unique Email
password	VARCHAR(255)	Encrypted Password

role	ENUM('ADMIN','CUSTOMER')	User Role

Products

Field	Type	Description
id	BIGINT	Primary Key
name	VARCHAR(200)	Product Name
description	TEXT	Product Description
price	DECIMAL(10,2)	Product Price
image_url	VARCHAR(500)	Image Link

Feedback

Field	Type	Description
id	BIGINT	Primary Key
user_id	BIGINT	Foreign Key (Users)
product_id	BIGINT	Foreign Key (Products)
rating	INT	1–5 Stars
comment	TEXT	User Comment
admin_comment	TEXT	Admin Response

---

7. Deployment Requirements

- Environment:
  - Java 17+
  - MySQL 8.0+
  - Apache Maven for build management
- Server Setup:
  - 4GB RAM, 2 CPU cores (minimum)
  - Deployed on Tomcat or embedded Spring Boot server

---

8. Legal and Compliance

- Must comply with data privacy and protection laws.

- Personal data (passwords, emails) must be securely stored and never shared.

---

9. Appendix

9.1 Assumptions

- Users have stable internet access.
- Database and backend are configured correctly.
- System clock synchronized for accurate timestamps.

9.2 Acceptance Criteria

- All Functional and Non-Functional requirements are implemented and tested.
- System performs under expected load.
- UI is responsive and easy to use.
- All data is securely handled and protected.

Version: 1.0
